IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 22 January 2024, accepted 29 January 2024, date of publication 2 February 2024, date of current version 16 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3361476

RESEARCH ARTICLE

Improving Quality of Service and HTTPS DDoS Detection in MEC Environment With a Cyber Deception-Based Architecture

JOELLE KABDJOU AND NORIHIKO SHINOMIYA, (Member, IEEE)
Graduate School of Science and Engineering, Soka University, Tokyo 192-8577, Japan
Corresponding authors: Joelle Kabdjou (e23D5351@soka-u.jp) and Norihiko Shinomiya (shinomi@ieee.org)

ABSTRACT Creating a cyber deception framework for 5G networks, particularly in IoT and cellular applications, is complex due to critical constraints in managing resources, meeting low latency demands, and addressing security concerns. While cloud computing aids in alleviating some limitations, it often falls short in meeting low-latency requirements. Multi-Access Edge Computing (MEC) has emerged as a solution by bringing resources closer to User Equipment (UEs) to reduce latency. Various MEC architectures have leveraged Software Defined Networking (SDN), Network Function Virtualization (NFV), Service Function Chaining (SFC), Network Slicing (NS), decision-making systems, and deception components. However, none have integrated these technologies comprehensively to achieve superior Quality of Service (QoS) and strengthen security. In this paper, we unify SDN, NFV, SFC, NS, decision-making technologies, and deception to efficiently manage MEC server resources and lure attackers. We utilize cyber deception metrics, including request collection rates over time and variations in request numbers concerning different botnet sizes. Moreover, we meticulously address QoS parameters such as latency, computing, storage, and bandwidth resources. Our approach initiates with a mathematical model for MEC server resource allocation, introducing a novel architecture that reduces bandwidth, computing, and storage resource usage. We introduce a cyber deception strategy utilizing uniform distribution and random selection to divert potential attackers. Simulations validate efficient resource management, notably reducing end-to-end latency for requests processed on the edge and in the cloud. This enhancement improves QoS within the MEC system and provides valuable insights for advancing decision-making technologies.

INDEX TERMS Multi-access edge computing (MEC), cyber deception, quality of service (QoS), software defined networking (SDN), network function virtualization (NFV), service function chaining (SFC), network slicing (NS), decision-making systems.

I. INTRODUCTION

Multi-access Edge Computing (MEC), introduced as a fundamental concept in the world of networking, represents a significant paradigm shift in how data processing and computing are distributed in modern networks. With the convergence of IoT and the rollout of 5G networks reshaping digital landscapes, MEC stands at the forefront of this transformation. It reduces the workload of network devices in applications such as 5G and 6G networks, IoT, augmented reality, and big data [1]. The MEC architectural shift is pivotal in enabling swift, high-capacity services with minimal latency. Positioned strategically at the network edge, MEC servers assume a critical role in data processing and storage, offering a diverse array of computing resources essential for the multifaceted functionalities of modern networks. By intelligently distributing computational loads from centralized cloud servers, MEC notably diminishes latency in crucial applications such as video streaming, IoT frameworks, and augmented reality. Moreover, this

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Yan.

approach optimizes bandwidth usage by processing data near end-users, amplifying the overall operational efficiency of networks [2]. Leveraging MEC's localized computational prowess, edge devices gain the capability to execute intricate computations, facilitating the deployment of real-time decision-making frameworks across diverse application domains. This transformative potential reshapes communication networks, fostering adaptive, responsive, and highly efficient ecosystems, marking the advent of a new era in connectivity and resource utilization. However, the proliferation of MEC also brings forth a new set of challenges, notably in the realm of security. As MEC continues to gain prominence, understanding its significance in the context of 5G and IoT while addressing the complex security challenges it presents becomes imperative for ensuring the resilience and reliability of the next generation of networked systems [3]. Security concerns have taken center stage in the realm of MEC, largely driven by the proliferation of connected devices and the emergence of novel applications. The shift toward edge computing, as articulated in [4], has introduced a fresh attack landscape, demanding a comprehensive scrutiny of security vulnerabilities inherent to MEC ecosystems. Particularly, Distributed Denial of Service (DDoS) attacks have gained prominence as a continually evolving menace. The exhaustive examination presented in [5] delves into the intricacies of DDoS threats, shedding light on their potential to disrupt services and undermine the overall user experience. In response to these pressing challenges, the MEC community has fervently committed to improving QoS, ensuring unwavering and reliable performance. Pioneering research initiatives, exemplified in [6], are diligently devising advanced resource allocation strategies aimed at optimizing QoS metrics while concurrently addressing security concerns. This collaborative effort marks a crucial integration of methods, strengthening MEC environments against new threats and improving the quality of user engagements.

Cyber deception techniques, strategically deploying elements like honeypots or decoy systems, are crucial in modern network security. Addressing MEC security challenges, they prove effective against threats such as DDoS attacks and unauthorized access. This deception redirects attackers and provides insights into their tactics. Past applications in network security have demonstrated their effectiveness, leveraging these techniques to fortify MEC security against evolving threats, enhancing resilience [7]. In [8], the author proposed that by integrating technologies such as SDN, NFV, SFC, and NS, it is possible to enhance QoS. Moreover, in [7], it was demonstrated that incorporating these technologies along with decision-making mechanisms can also bolster the security of MEC servers. The synergy of SDN and NFV allows for the separation of the data plane from the control plane, enabling rapid data processing. SFC leverages multiple NFV instances to enhance efficiency, while NS facilitates the use of specific network slices for real-time data forwarding. Numerous solutions have been put forth in the past, aiming to improve QoS and implement a deception framework for threat assessment. However, to the best of our knowledge, none of these solutions have comprehensively addressed the simultaneous enhancement of QoS, the management of deception-related threats, and the intricacies of the attacker-system interaction critical for decision-making in MEC networks. In this study, our motivation is twofold: first, to leverage the aforementioned technologies to elevate the quality of service within MEC frameworks, and second, to deceive potential attackers, particularly in the context of DDoS attacks. To achieve our goal, we commence by examining the existing body of research in the MEC domain. Within this field, various investigations have utilized technologies like SDN, NFV, SFC, NS, and decision-making methodologies, either independently or in different combinations, to enhance QoS within the MEC infrastructure. As we identify challenges inherent in these approaches, we propose a more robust solution that integrates all these technologies. Our approach involves a process for managing requests and responses within a framework of cyber deception. This framework's foundation relies on utilizing Uniform distribution and random strategy selection to govern the dynamics of interactions between potential attackers and the MEC system. The use of Uniform distribution ensures that each potential response has an equal chance of engaging with the MEC system, guaranteeing a fair and unbiased approach to security. Meanwhile, random strategy selection introduces an element of unpredictability, making it challenging for attackers to anticipate the response generated from the MEC server. Simulation results clearly demonstrate that our comprehensive solution not only ensures heightened QoS for MEC networks but also enables extensive data acquisition related to potential attackers.

The remaining sections of this paper are structured as follows: In Section II, we conduct a review of related works in the realm of QoS enhancement within the MEC framework, discuss various security measures against DDoS attacks, and present the underlying motivation for our research. Section III introduces our solution, which employs Cyber Deception to augment DDoS detection and mitigation in MEC, utilizing SDN, Decision-Maker Technology, SFC, NFV, and NS. Section IV provides insights into the results of our simulations. Finally, we conclude this paper in Section IV-E, where we also provide recommendations and address unresolved issues.

II. RELATED WORK

MEC is an architectural approach that extends cloud computing to the network edge, positioning computational power closer to data sources and utilization points. This proximity significantly minimizes latency, enhances QoS, and enables real-time data processing across diverse applications like IoT, augmented reality, and autonomous vehicles. As emphasized in [7], the importance of MEC in reducing latency, particularly in IoT and 5G technology, cannot be understated. MEC encounters several challenges, including task offloading, congestion control, resource allocation,

security, privacy, mobility, and standardization. In this paper, our primary focus is on security concerns. Security assumes paramount importance in MEC due to its operation at the network's edge, making it vulnerable to various cyber threats. Protecting sensitive data, fortifying edge devices, ensuring secure data transmission, and defending against threats like DDoS attacks are critical aspects, as elaborated in [9]. In [10], the authors introduced a security framework designed for 5G MEC networks, concentrating on enhancing secure access to network elements and introducing the MEC Enabler for efficient credential management. They explored various usage scenarios and provided an access control protocol diagram to illustrate the authentication process within MEC-driven services.

Cyber deception is a strategy that utilizes false information and decoys in networks to mislead and divert attackers. The goal is to lure, confuse, or delay attackers while simultaneously detecting their presence and gaining insights into their tactics, allowing for early threat detection and enhanced cybersecurity defenses [11]. Furthermore, in [12], the authors addressed the critical issue of DDoS attacks on SDN IoT-Edge Computing, exacerbated by the pandemic-induced remote work trend. They explore the efficacy of an SDN-based Moving Target Defense (MTD) technique within a smart building context. The study's MTD Reactive and Proactive Network Address Shuffling Mechanism successfully defends against UDP, TCP SYN, and LAND DDoS attacks, safeguarding IoT devices from botnet compromise by frequently changing network addresses while ensuring reliable system performance. Additionally, in [13], a system designed to identify and prevent DDoS attack traffic stemming from compromised IoT devices is introduced. This is achieved by monitoring and analyzing specific packet types—TCP, SYN, ICMP, and DNS—originating from these IoT devices. Cyber deception enhances DDoS detection by providing early warnings via decoys, reducing false positives, conserving network resources, segmenting attacks, obfuscating attacker intent, providing insights into malicious behaviors, and enriching threat intelligence [14]. In [7], authors introduced a deceptive MEC architecture, integrating NFV, SFC, SDN, NS, and decision-maker technology to combat cyber threats. They assessed its efficacy through Monte Carlo simulations, demonstrating its effectiveness in enhancing security. However, the Quality of Service (QoS) remains a critical issue in MEC scenarios, involving the employment of mechanisms or technologies within a network to regulate traffic and ensure optimal performance for essential applications when network resources are limited. Moreover, in [15], notable contributions were made to enhancing QoS within the context of MEC by integrating SDN, NFV, SFC, and NS technologies to efficiently oversee MEC server resources, ensuring reliable QoS requirements for AVNET are met. Additionally, in [16], a method aimed at improving both data privacy and security to enhance the Quality of Experience (QoE) within a MEC environment was introduced. The key contribution of this work is the proposal of a hybrid cryptography system, combining both symmetric and asymmetric cryptography techniques, to enhance data security, privacy, and user authentication within a MEC-based network. Notably, none of the previously mentioned solutions ([7], [15], [16]) address the dual aspects of improving QoS and enhancing user interaction with the MEC computing server within the deceptive framework. To the best of our knowledge, this pioneering work comprehensively addresses both these facets simultaneously.

Our objective is to introduce an MEC solution that harnesses SDN, NFV, SFC, NS, and decision-making technologies to protect MEC servers from DDoS attacks. This solution considers resource limitations such as latency, processing power, storage, and bandwidth, all while aiming to meet QoS requirements in MEC environments. It also evaluates the volume of data collected from attackers and the methodology used for its accumulation over time. These endeavors are geared toward enhancing decision-making technology within the system.

III. CYBER DECEPTION FOR ENHANCED DDoS DETECTION AND MITIGATION IN MEC: LEVERAGING SDN, DECISION-MAKING TECHNOLOGY, SFC, NFV, AND NS

As proposed by both [8] and [7], the integration of diverse technologies to enhance QoS and deception tactics within the MEC framework has been a consistent suggestion. Expanding upon this advice, we introduce an innovative MEC solution in this paper. Our approach harnesses an extensive array of technologies—such as SDN, NFV, NS, SFC, and decision-making technology—not only to elevate QoS but also to lure attackers and gather data about them.

A. PROPOSED ARCHITECTURAL FRAMEWORK

Our presented architectural framework is visually represented in Figure 1, designed to facilitate the mobility of end devices while establishing crucial communication links to prevent collisions with other devices. End devices achieve this through direct communication or by sending requests to the nearest Base Station (BS), which, in turn, connects them to the MEC servers. Upon receiving a user's request, the MEC server follows the prescribed procedures outlined in Sections III-B and III-D.

Our innovative solution is meticulously crafted to ensure minimal latency, efficient resource management encompassing bandwidth, computing, and storage, enhance ambiguity for malicious devices, and compile valuable information about potential attackers to support the decision-making system. This multifaceted approach is realized through the integration of cutting-edge technologies, including SDN, NFV, SFC, NS, deception component and decision-making. MEC servers maintain open lines of communication with a diverse array of network components, such as cloud data centers, end devices, and other MEC servers, fostering these connections directly or through the BSs. At the heart of our architectural design is the strategic inclusion of a deception

component and the adept management of request/response interactions within the internal structure of the MEC server.

B. REQUEST PROCESS MANAGEMENT

- In this architectural setup, four distinct device categories are identified: end devices encompassing both normal and malicious entities. When engaging with the MEC server's information or services, end users may delegate tasks to it. Malicious users may undertake harmful actions resulting in adverse consequences such as performance deterioration, resource depletion, server overload, and security vulnerabilities. Conversely, normal users may aim to execute legitimate tasks on the MEC server due to resource limitations on their devices. This situation arises due to the widespread use of IoT, with numerous devices lacking computing resources and relying on the MEC server for computation. The test scenario in this paper centers around subjecting the MEC server to DDoS attacks using the GET and POST methods.
- As outlined by the author in [17], the MEC server is partitioned into five virtual machines to address specific requirements, notably assuring QoS and user experience. VM1 hosts both the Decision Maker and the SDN controller. Incoming requests directed at the MEC server are initially intercepted by the SDN controller, possessing a comprehensive understanding of the entire MEC server architecture and maintaining communication with the virtual machines [18]. Upon reception, the SDN controller relays the request to the Decision Maker, tasked with determining the request's legitimacy (i.e., identifying potential DDoS attacks) and subsequently transmitting the findings back to the SDN controller for further assessment. It is presumed that a botnet and the decision maker's view have been established. Once the SDN controller receives the Decision Maker's response:
  - If the response confirms the request's legitimacy, the SDN controller routes the request to the VNF checker (3). The VNF checker assesses whether the available resources within the MEC server can accommodate the request and furnishes a response to the SDN controller (4).
    * Upon receiving a "Yes" response from the checker, indicating ample resources within the MEC server, the SDN controller forwards the request to the VNF processor (10). The VNF processor carries out the computation and transmits the response back to the SDN controller (11). The received response is then dispatched to the VNF receiver (VM3) for storage in the Cloud data center (8) and delivery to the legitimate device.
    * If the checker's response indicates "No," signifying insufficient resources, the SDN controller initiates an alternative course of action. It directs the request to the VNF sender (VM3) (1), which then transfers the request to the adjacent MEC server directly connected to the specific MEC server where the VNF sender is integrated (7).
  - When the Decision Maker's response identifies a DDoS request, the SDN controller routes the request to the VNF deception processor (VM5) for in-depth analysis and establishment of the deception interaction procedure (5 and 6). In this context, two scenarios arise:
    * If the database (DB) in the VNF deception processor is empty, indicating no recorded DDoS GET requests, the deception processor conveys the request to the SDN controller, which forwards it to the VNF processor for further scrutiny. The analysis involves generating an augmented reality version of the requested video, as described in [20] for fake video generation. We assume that the decision maker's strategy for HTTPS GET and POST DDoS attacks detection has been previously described in [21]. Subsequently, the SDN controller relays the analysis results to the deception processor. The deception processor archives this information in the DB for future reference, using either randomization or uniform distribution to select a status code for the GET response. This response is transmitted to the SDN controller, which forwards it to the malicious device.
    * If the DB within the VNF deception processor is not empty, upon receiving the request from the SDN controller, the processor checks if there is any correspondence within the DB corresponding to the specific user. If such correspondence is identified, the processor applies the previously described scenario until the request received from the malicious device no longer matches the one stored in the DB. Regarding the POST request, a similar process is implemented, except that the request is sent to the VNF processor for analysis, as a POST request pertains to an upload process. The handling process is detailed in Algorithm 1.
- The cloud server plays a pivotal role in data storage, housing the information and analytical outcomes produced by the MEC server. It offers extensive storage capabilities, serving as a potential redundancy system for MEC Edge Servers.
- The VNF deception processor serves as the mechanism for redirecting malicious traffic. Within this architectural framework, the SDN controller functions as a gateway, rerouting unauthorized inbound traffic towards the deception MEC server. This component accumulates extensive data on potential attackers, leveraged for

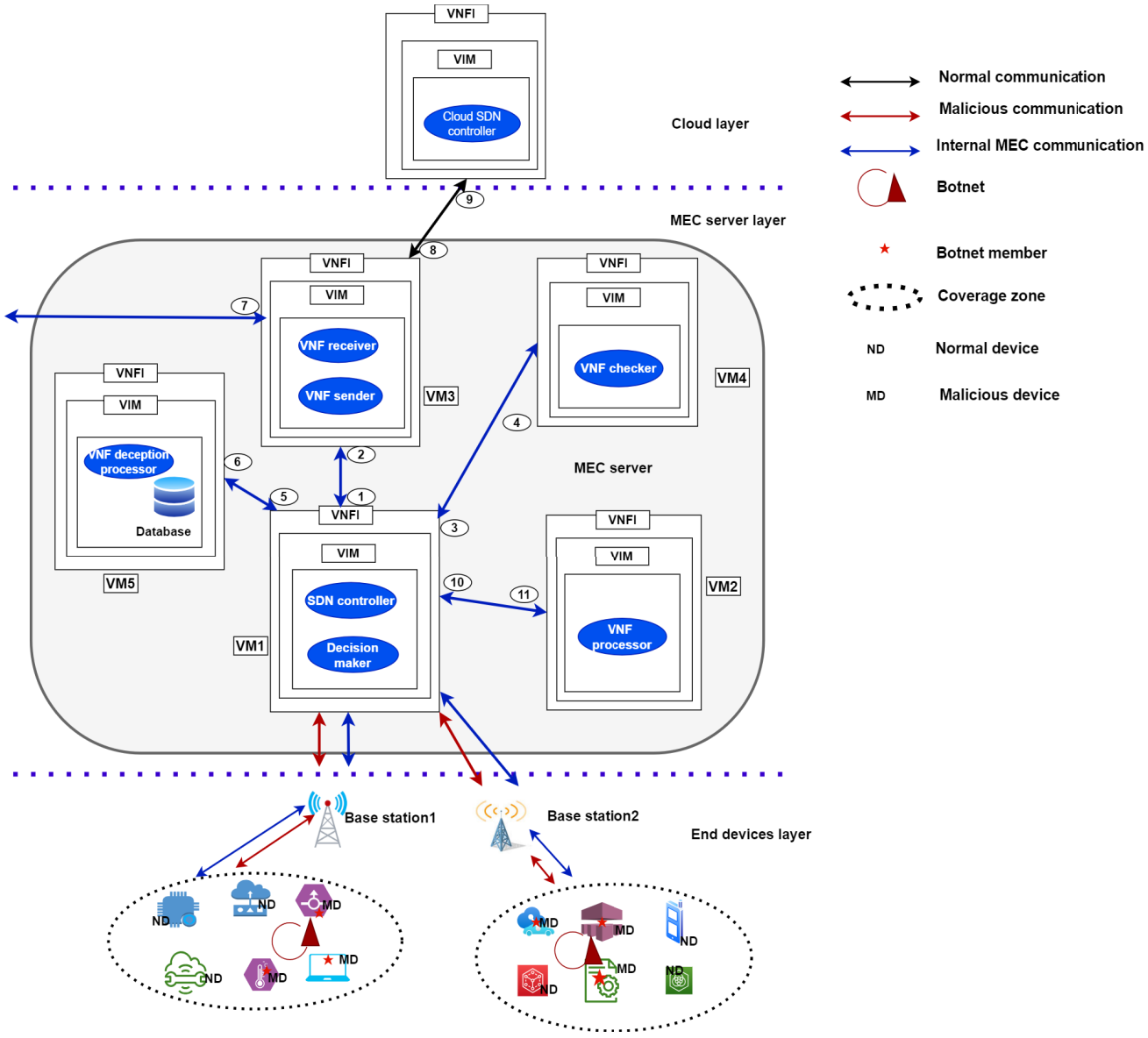


FIGURE 1. Our proposed MEC architecture.

decision-making within the MEC server. The aim is to diminish future attacks within the MEC server.

1) CYBER DECEPTION STRATEGY

Our deception strategy involves the application of either the Uniform distribution or Randomized strategy. These employ a predefined set of status codes to formulate responses following a DDoS attack on the MEC server. The status codes represent the response options utilized by the deception processor, with the choice between the Uniform distribution approach and the Randomized strategy approach determining their application. A comprehensive list of the status codes utilized is provided in Table 2.

2) UNIFORM DISTRIBUTION APPROACH

The interaction involves the MEC server representing the system, and the attacker acting as the malicious device. The primary objective of the MEC server is to generate responses that are indistinguishable as fake by the attacker. To achieve this goal, a specific strategy has been deployed: the initial step involves the implementation of a Uniform distribution approach among the response categories, each associated with distinct status codes. This approach guarantees an even distribution of response selections across various status codes, amplifying the appearance of randomness. Consequently, this makes it more difficult for the attacker to differentiate between authentic and counterfeit responses.

TABLE 1. Input values for algorithm 1.

Input	Description
LT	List of tasks request
LT_f	List of illegitimate tasks
LT_t	List of legitimate tasks
E_i	MEC server with id i
D_j	End device with id j
$D_{j_{pos}}$	End device position
CR_f	Computing resources for tasks in LT_f
CR_t	Computing resources for tasks in LT_t
SR_f	Storage resources for tasks in LT_f
SR_t	Storage resources for tasks in LT_t
$T^t_{E_i}$	Task request received by E_i at time t
$T^t_{jE_i}$	Task request received by E_i at time t from D_j
$T^t_{jtE_i}$	Task request received by E_i at time t from normal device D_j
$T^t_{jfE_i}$	Task request received by E_i at time t from malicious device D_j
C_{E_i}	Computing resources in E_i
S_{E_i}	Storage resources in E_i
θ_{E_i}	Internal tasks bandwidth requests in E_i
δ_{E_i}	Storage resources for internal tasks in E_i
$\lambda_{T^t_{jfE_i}}$	Storage resources for $T^t_{jfE_i}$ at time t
$\lambda_{T^t_{jtE_i}}$	Storage resources for $T^t_{jtE_i}$ at time t
$\Omega_{T^t_{jtE_i}}$	Computing resources for $T^t_{jtE_i}$ at time t
$\Omega_{T^t_{jfE_i}}$	Computing resources for $T^t_{jfE_i}$ at time t
$R_{T^t_{jE_i}}$	Response received by D_j at time t from E_i

3) RANDOMIZE APPROACH

Regarding the Randomize strategy approach, each response has a random chance of being chosen among the set of responses. This selection method enhances the attacker's uncertainty. To simulate the scenario in which the MEC server randomly selects a response from a set of responses to send to an attacker, depending on the GET or POST request.

4) ATTACKER AND DECEPTION PROCESSOR VIEW

The scenario outlined in this paper enables users to engage in both data downloading and uploading activities, primarily involving video content. This step presumes that the trustworthiness of the request has already been established by the Decision Maker, determining the presence of DDoS attacks. The subsequent handling process is articulated as follows:

- VNF deception view:
  - -- In the context of GET requests, the deception processor initiates a validation process to ascertain the status of the user's session. Upon detecting an active session and additional request data in the local database, the server presents two distinct approaches for consideration: the randomize approach and the uniform distribution approach. Subsequently, the server adjusts the status code within the corresponding response before formally crafting and transmitting it back to the user. Simultaneously, the requested data is relayed to the VNF processor, facilitating augmented reality computations within the specified video and thereby introducing an element of heightened uncertainty for potential attackers. If the user's session has expired, access is temporarily restricted until the initiation of a new session.

Algorithm 1 Storage and Computing Resources Handling Process

Input : Table 1
Output: $R_{T^t_{jfE_i}}$, $R_{T^t_{jtE_i}}$

Insert $T^t_{jE_i}$ in LT
$R_{T^{jtE_i}_t} \leftarrow null$
$R_{T^{jfE_i}_t} \leftarrow null$
$R_{T^{jE_i}_t} \leftarrow null$
$CR_{E_i} \leftarrow CR_{E_i} + \Omega_{T^t_{jE_i}}$
$SR_{E_i} \leftarrow SR_{E_i} + \lambda_{T^t_{jE_i}}$
$D_{j_{pos}} \leftarrow CheckPosition(D_j, E_i)$
$R_{filterT^t_{jE_i}} \leftarrow Filter(T^t_{jE_i})$
if $R_{filterT^t_{jE_i}} = false$ **then**
  Insert $T^t_{jE_i}$ in LT_f
  $T^t_{jfE_i} \leftarrow T^t_{jE_i}$
  if *GET request* **then**
    $T^t_{jfE_i} \leftarrow$ Sent-Resp$(T^t_{jfE_i}, Dj)$
    sent to deception processor
  else
    sent to deception processor
  Store $T^t_{jfE_i}$ in the deception database
else
  Insert $T^t_{jE_i}$ in LT_t
  $T^t_{jlE_i} \leftarrow T^t_{jE_i}$
  if $D_{j_{pos}}$ *is coverage zone* (E_i) **then**
    if $CR_t \leq C_{E_i} - \theta_{E_i}, SR_t \leq S_{E_i} - \delta_{E_i}$ **then**
      $R_{T^t_{jE_i}} \leftarrow compute(T^t_{jE_i})$
    else
      Forward $T^t_{jE_i}$ to the cloud server and wait for the response
      while $R_{T^t_{jE_i}} = null$ **do**
      $R_{T^t_{jE_i}} \leftarrow Result - cloud - server(T^t_{jE_i})$
      $T^t_{jtE_i} \leftarrow T^t_{jE_i}$
    Forward $R_{T^t_{jtE_i}}$ to D_j
    Store $R_{T^t_{jtE_i}}$ to cloud server
  else
    Forward $T^t_{jtE_i}$ to the next MEC server
$CR_f \leftarrow CR_f + \Omega_{T^t_{jfE_i}}$
$SR_f \leftarrow SR_f + \lambda_{T^t_{jfE_i}}$
$CR_t \leftarrow CR_t + \Omega_{T^t_{jtE_i}}$
$SR_t \leftarrow SR_t + \lambda_{T^t_{jtE_i}}$
remove $T^t_{jtE_i}$ from LT_t
remove $T^t_{jfE_i}$ from LT_f
remove $T^t_{jE_i}$ from LT

TABLE 2. Frequent status code used for GET, and POST https requests.

GET	POST
200, 301, 302,304, 400, 401,	200, 201, 204, 400,401,403,
403, 404, 502, 503, 504	405, 409, 500,502, 503, 504

- For POST requests, a process similar to that of GET requests is implemented, with a notable exception: the deception processor refrains from forwarding any requests to the VNF processor. Instead, the deception server is solely dedicated to structuring the response with the requisite fields and determining the status code—a decision governed by either the randomization or uniform distribution method.

- Attacker's view:
  In this situation, malicious actors implement a DDoS tactic involving the indiscriminate flood of requests originating from various user sources. These attackers work with a predetermined number of requests to release. On the receiving side, users continuously initiate requests, pursuing their intended goals, which might entail obtaining a status code of 200 for POST requests or successfully retrieving the specific data they've requested in the case of GET requests. In this context, the structure of the requests plays a crucial role, encompassing essential elements such as header details, content, status codes, and other vital particulars.

C. BASELINE WORKS

The referenced baselines within this study delineate two distinct approaches: one devoid of decision-making technology and deception, and the other lacking an MEC server:

- Approach Without Decision-Making Technology and Deception: This methodology utilizes the internal MEC architecture depicted in Figure 1 but excludes both the deception Network Function Infrastructure (NFI) and decision-making technology. Consequently, there exists no filtration mechanism for end-device requests. All incoming requests are processed either within the current server or rerouted to the subsequent MEC server if the user's location falls outside the coverage zone of the current server. In cases where the end device is within the server's coverage but local resources in the MEC are inadequate to handle the request, computation occurs in the cloud server, as outlined in [15].
- Approach Without MEC Server: This approach represents the conventional cloud setup devoid of an MEC server in close proximity to the data producer. In this scenario, all user requests undergo processing exclusively within the cloud server.

D. SFC GRAPH AND NETWORK SLICING IN THE PROPOSED ARCHITECTURE

1) SFC GRAPH

Service Function Chaining (SFC) stands out as a vital technology that serves to decompose complex network services, known for their resource demands, into a series of interconnected virtual network functions [22]. This integration carries substantial benefits, primarily marked by a reduction in end-to-end latency—a central QoS metric that we emphasize. When woven into the fabric of SDN and NFV technologies, SFC excels at facilitating the efficient orchestration and deployment of service functions. This empowers the categorization and enforcement of policies, allowing for the routing of data flows based on specific service requirements and the current network status. Ensuring the streamlined provisioning of SFC requests holds paramount importance, particularly in facilitating the operation of ultra-low latency applications while minimizing the consumption of physical resources. Telecommunication service providers distinctly favor the optimization of existing physical network resources—bandwidth, CPU, and RAM memory—within the network architecture over the acquisition of supplementary physical network resources. For a more visual representation, refer to Figure 2 depicting the SFC architecture within our MEC framework.

As highlighted by the authors in [23], an SFC graph essentially comprises two integral components: the SFC data plane (SFC-DP) and the SFC control plane (SFC-CP). These two facets are interconnected through four distinct interfaces, although in our specific case, only three of these interfaces are relevant:

- Interface C1, linking SFC-CP and SFC-Cl, is primarily designated for the management of SFC classification rules within classifiers or decision-making components.
- Interface C2, facilitating communication between SFC-CP and SFF, is primarily employed to exchange necessary data related to SFC forwarding decision-making, as well as to collect state information concerning SFPs and other relevant details.
- Interface C3 serves as the bridge between SFC-CP and SFC-aware SF and is chiefly used for functions such as gathering output data generated during packet processing within the SF.

Our MEC architecture integrates the SDN controller, which assumes the role of the SFC controller within the proposed SFC graph. It is situated inside VM1, as illustrated in Figure 1. Additionally, the SFC Classifier functions as a VNF that resides in VM1. Its primary responsibility in our architectural framework involves filtering incoming traffic originating from the SFC controller and then, based on the type and legitimacy of the request, directing it to SFF1, SFF2, or SFF3. When the task originates from the flow end devices or VM3 in Figure 2, the SFC Classifier labels the flow for routing to SFF1 or SFF2. In contrast, tasks originating from flow 3 are tagged by the SFC Classifier for forwarding to SFF3.

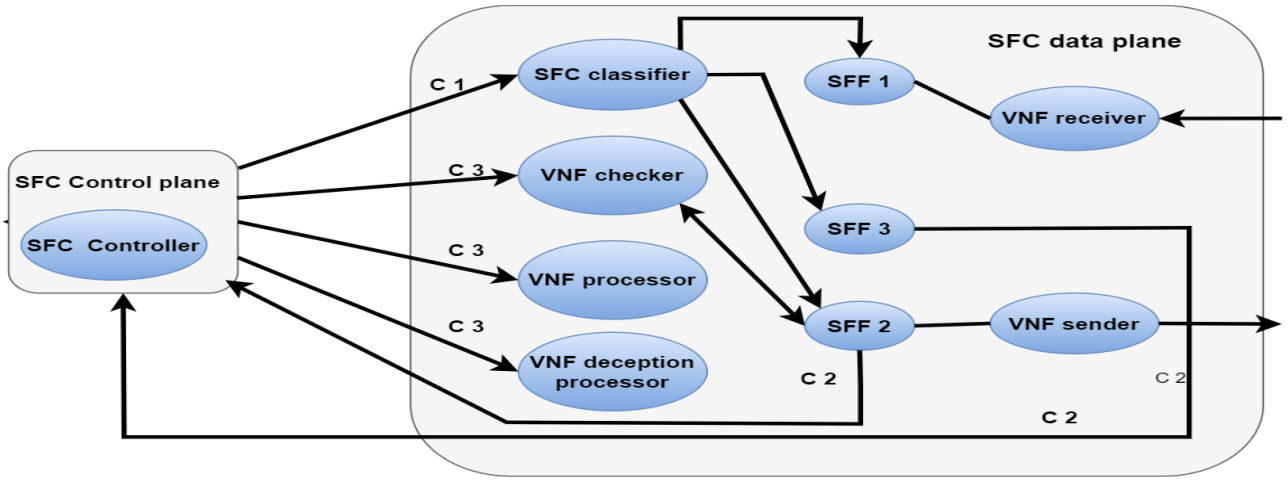


FIGURE 2. Our service function chain process handling.

Our SFC graph comprises three SFFs: SFF1, SFF2, and SFF3, respectively situated in VM1, VM4, and VM5. When SFF1 receives a task from the SFC Classifier, it forwards it using the VNF Sender function. If the task reaches the VNF Receiver, it is subsequently directed to SFF2. In the case of illegitimate tasks, after filtration, they are forwarded to SFF3.

Within SFF2, tasks arriving from SFF1 are immediately forwarded to the SFC controller via interface C2. Conversely, when tasks arrive from the SFC classifier, the SFF2 makes routing decisions based on task parameters, subsequently directing them either to the VNF Checker or the VNF Processor. Upon processing, the VNF Checker/Receiver transmits the results to the SFC controller through interface C3.

Finally, tasks received by the SDN controller via interface C2 are relayed to the SFC Classifier using interface C1. When tasks are sourced from C3, the SFC controller performs a check to determine if they originate from the VNF Processor or the VNF Checker. In the former case, the SFC controller issues a response to the requesting end device, while in the latter case, the task is routed to the SFC Classifier through interface C1. VM4 utilizes interface C2 to communicate with the SDN controller.

2) NETWORK SLICING APPROACH

Network slicing, a concept in network architecture, involves dividing a single physical network infrastructure into multiple isolated virtual networks or "slices," each customized for specific applications or services. These slices provide tailored resources like bandwidth, processing power, and security features to suit diverse use cases such as IoT, autonomous vehicles, and ultra-low-latency communication. It's a fundamental part of 5G networks, enabling efficient resource allocation, end-to-end control, and flexibility in service delivery for optimal performance and security.

Leveraging network slicing technology within an MEC-SDN-NFV-Decision Maker architecture is crucial for enhancing QoS. However, it comes with challenges such as interoperability, mobility considerations, orchestration efficiency, and tailored business models, depending on specific application needs [24]. The cloud SDN controller holds a pivotal role in this, responsible for establishing logical resource slicing and meeting QoS criteria for each slice, as outlined in [25]. It dynamically adjusts QoS parameters based on unique slice requirements, communicating decisions to the MEC SDN controller for physical resource management and user scheduling.

The network slicing approach here aligns with [15] requirements, delineating communication channels among various components: end devices, end devices-MEC servers, MEC servers-cloud data centers, and interconnections between MEC servers. This strategy introduces two specialized slices: the low-latency slice for rapid communication between end devices and MEC servers and the high-latency slice for MEC servers' communication with cloud servers. Mobility considerations for end devices are accounted for in the low-latency slice, allowing for flexible path alterations based on specified proba bilities [15]. We employed the Manhattan mobility model for simplicity in depicting these movements.

In contrast, the high-latency slice handles communication between MEC and cloud servers, optimized for higher latency without compromising connectivity strength. Actions include storing computation results on cloud servers (as indicated in line 30 of Algorithm 1) or transmitting network maps between cloud and MEC servers. These interactions rely on the bandwidth fraction γ introduced in Section III-E2. The allocation of this fraction is managed by the cloud SDN controller based on overall network traffic volume, ensuring secure data transfers.

E. RESOURCE MANAGEMENT FOR COMPUTING AND STORAGE

1) COMPUTING AND STORAGE MANAGEMENT

In a network setup, there's a cloud data center hosting a single server denoted as C, several MEC servers labeled as

TABLE 3. Notation list.

symbols	Description
m	number of MEC servers
β	Total available bandwidth in the network
R	Network radius surface
$E_i(1 \leq i \leq m)$	The MEC server with identification i
$C_i(1 \leq i \leq m)$	Computing resources for E_i
$S_i(1 \leq i \leq m)$	Storage resources for E_i
r_i	E_i radius surface
n	Number of end devices
$D_i(1 \leq i \leq n)$	end device with id = i
$T^t D_{iE_j}$	Request send by Di to E_j at time t
$\alpha^t D_{iE_j}$	Require computing resources for the request send by D_i at time t
$\lambda T^t D_{iE_j}$	Require storage resources for the request send by D_i at time t
b $T^t D_{iE_j}$	Require bandwidth to compute the request send by D_i at time t
$d^t D_i$	Direction of Di at time t
ρD_i	velocity of D_i
θE_i	Internal task computing resources inE_i
s_{E_i}	Internal tasks storage resources in E_i
γ	Non delay-sensitive data bandwidth

E_i (where $1 \leq i \leq m$), and a set of end devices termed D_i (where $1 \leq i \leq n$). The network provides a total available bandwidth β within a radius R. Each MEC server, E_i, has its computing resources (C_i), storage capacity (S_i), and a unique coverage zone defined by r_i.

In this network structure, the notation $T^t_{DiE_i}$ represents a request initiated by end device D_i directed towards MEC server E_i at a specific time, denoted as t.

The goal of managing computing and storage resources encompasses two primary objectives. First, it aims to ensure prompt processing of incoming requests to the MEC server. Second, it aims to efficiently transmit computation outcomes back to the requesting end device, minimizing latency. To achieve these goals, the solution needs to route tasks to the nearest MEC server with adequate resources for computation and storage. Within MEC server E_j, two crucial parameters come into play: θ_{E_i} and s_{E_i}, representing the internal computing and storage resources required to process and store requests not directly associated with end device processing and communication respectively. The available computing and storage resources within the MEC server must exceed or equal the cumulative resource requirements of all processed tasks, encompassing both external requests and internal operations. Additionally, computation can only commence if the originating end device falls within the coverage radius of the MEC server. It can be observed that integrating the decision maker into this architecture results in a reduction of both computing and storage resources, in contrast to an architecture lacking request filtering within the internal structure of the MEC server. Now, assuming there are k legitimate requests and q illegitimate requests denoted as T^m for malicious requests and T^n for normal requests, the allocation of computing resources, as expressed in Equation 1, needs to consider that only GET requests directed to the MEC server for data download require significant computing resources. Since we've assigned this computational task to the VNF processor, it demands substantial computing resources. Conversely, for POST requests, the complexity primarily involves formalization and can be approximated with a constant time complexity.

$$\sum_{z=1}^{k} \alpha_{T^n{}_z} + \sum_{p=1}^{q} \alpha_{T^m{}_p} \leq C_{E_i} + \theta_{E_i} \tag{1}$$

$$\sum_{z=1}^{k} \alpha_{T^n_z} + \sum_{z=1}^{q} \alpha_{T^m_z} \leq \sum_{z=1}^{k} \alpha_{T^n_z} + \sum_{z=1}^{q} \alpha_{T^n_z} \tag{2}$$

$$\sum_{z=1}^{q} \alpha_{T^m_z} \leq \sum_{z=1}^{q} \alpha_{T^n_z}$$

$$\rightarrow \sum_{z=1}^{q} \alpha_{T^m_z} + \sum_{z=1}^{k} \alpha_{T^n_z} \leq \sum_{z=1}^{q} \alpha_{T^n_z} + \sum_{z=1}^{k} \alpha_{T^n_z} \tag{3}$$

The same process is applied to the storage resources, and we have observed a similar pattern of efficient utilization.

$$\sum_{z=1}^{q} \lambda_{T^m_z} \leq \sum_{z=1}^{q} \lambda_{T^n_z} \rightarrow \sum_{z=1}^{q} \lambda_{T^m_z} + \sum_{z=1}^{k} \lambda_{T^n_z} \leq \sum_{z=1}^{q} \lambda_{T^n_z} + \sum_{z=1}^{k} \lambda_{T^n_z} \tag{4}$$

2) BANDWIDTH MANAGEMENT

In our network slicing approach, effective bandwidth management is crucial to ensure that each slice has a sufficient share of bandwidth resources to handle its respective request load, meeting the QoS requirements, especially low latency. Equation 5 introduces a bandwidth constraint by considering the total available bandwidth, denoted as β, and the bandwidth required to process each incoming request.

$$\sum_{i=1}^{m} \sum_{j=1}^{n_i} b_j \leq \gamma + \beta \tag{5}$$

Here, m represents the count of MEC servers, n symbolizes the number of tasks assigned to MEC server E_i, and b_j signifies the bandwidth requirement for task T_j, specifically for ensuring a low-latency transmission of the processing results. Additionally, γ denotes the bandwidth essential for facilitating data exchange between MEC servers and the cloud server, particularly for transmitting non time-sensitive data, such as storing task results in the cloud server or sharing network map updates with MEC servers.

The allocation of bandwidth is proportional to the quantity of requests each MEC server must handle. To avoid situations where some MEC servers have unused allocated bandwidth while others struggle due to resource limitations, the Cloud SDN Controller employs a predictive machine learning model. This model anticipates the optimal allocation of bandwidth resources for each MEC server by leveraging data on end device movements over a specific time interval. The prediction model utilizes lists of MEC servers and stored

results to forecast the future flow of end devices. This prediction process utilizes link quality estimation techniques outlined in [26]. Based on these predictions, bandwidth allocation is carried out for each MEC server, considering the parameter γ to comply with the constraint specified in Equation 5.

F. CYBER DECEPTION STRATEGY: IMPACT ON NETWORK SECURITY

Cyber deception involves deploying deceptive elements like decoys, false data, and altered environments to mislead attackers, diverting unauthorized access attempts. Implementing a cyber deception strategy has multifaceted implications for network security, covering crucial aspects:

- It introduces proactive measures that actively mislead and redirect potential attackers, bolstering existing security measures and reducing the network's vulnerability to a wide range of cyber threats [27].
- Deception tactics effectively mitigate advanced threats, diverting assailants away from critical assets and into fabricated environments, thus safeguarding against targeted intrusions and sophisticated malware.
- By luring attackers into false environments, organizations gain invaluable insights into adversarial tactics, fostering a deeper understanding of evolving cyber threats and enabling proactive adjustments in defense strategies.
- The proactive nature of cyber deception aids in early threat detection, reducing dwell time within the network. It provides opportunities to neutralize potential breaches before they escalate, averting data compromises [29].
- Ultimately, cyber deception strengthens network resilience by misleading adversaries, minimizing the impact of successful cyberattacks, and enabling swift recovery measures to maintain network integrity and functionality. This is particularly crucial in mitigating threats like DDoS attacks, a primary concern addressed in this work.

IV. SIMULATION RESULTS

In this section, we present the results and analysis of simulations conducted to assess the efficacy of our proposed cyber deception-based architecture. These simulations were executed using the EdgeCloudSim simulator, with an HTTPS service installed via Python scripts to launch GET and POST DDoS requests. EdgeCloudSim is an open-source simulation platform tailored for Edge Computing scenarios, enabling experimentation involving both computational and network resources. Windows 11 served as the operating system for these simulations. A comprehensive overview of the parameter values used in the simulations is provided in Table 4.

The primary objective of these simulations is to demonstrate that our architecture, which integrates a decision-making approach, leads to more efficient utilization of network resources compared to scenarios lacking decision-making technology. Additionally, the simulations illustrate how our approach facilitates information collection about malicious devices while enhancing their level of uncertainty.

TABLE 4. Setting parameters.

Settings	Values
Number of servers MEC	4
Number of Cloud server	1
wan_bandwidth	1000
man_bandwidth	500
wlan_bandwidth	200
lan_bandwidth	15
computing_resources	100000 MIPS
storing_resources	10000 MB
ram_resources	10240 MB
min_av_distance_to_mec_server	25
max_av_distance_to_mec_server	1000
mec_server_radius_coverage	500
min_number_of_mobile_devices	10
max_number_of_mobile_devices	50
Direct communication latency	1.95
botnet size	300
Maximum size of downloaded video	300 kb
Maximum size of uploaded video	1000kb
Session length	180 s

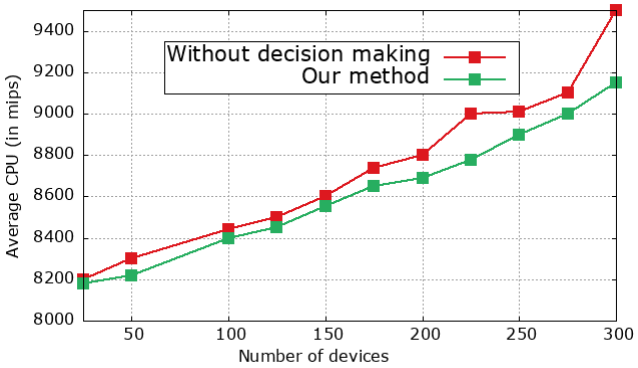


FIGURE 3. The computing resources evaluation.

A. COMPUTING AND STORAGE RESOURCES EVALUATION

To evaluate the efficiency of Algorithm 1 concerning computing and storage resources, we analyzed resource utilization in two scenarios: one integrating decision-making technology and another without it. In the approach lacking a decision-making system, no filtration process or deception component is employed. All requests are handled by solely verifying resource availability on the MEC server. In Figure 3, our focus was on CPU resource assessment. We observed that with an increasing number of end devices, the average computing resources also rose. This increase can be attributed to the fact that the decision-making-absent approach involved numerous operations before responding to both DDoS requests and legitimate requests, resulting in higher CPU usage. For instance, with 250 devices, the average computing resources used were 8900 MIPS and 9001 MIPS for the methods with and without decision-making, respectively.

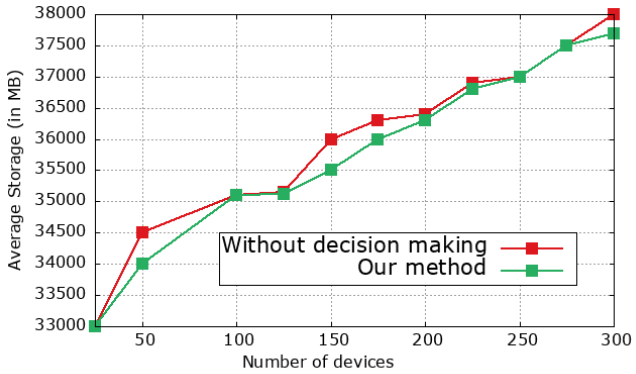


FIGURE 4. The storage resources evaluation.

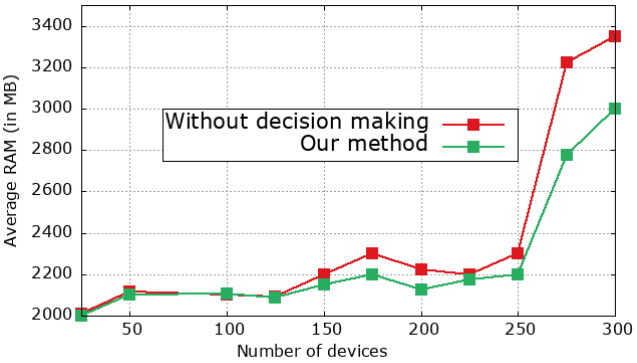


FIGURE 5. Average RAM used.

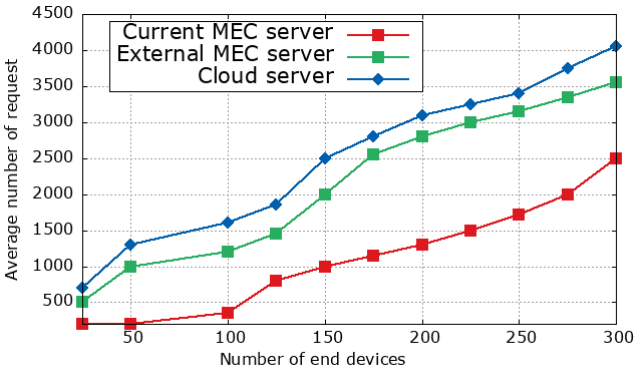


FIGURE 6. Analyzing variations in request processing locations.

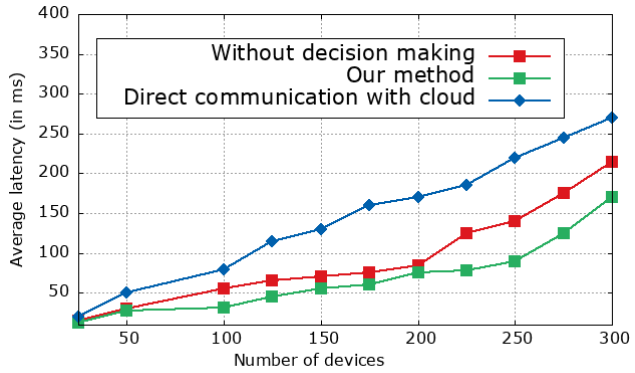


FIGURE 7. The latency evaluation.

Furthermore, we assessed RAM utilization as depicted in Figure 4. Interestingly, unlike the CPU evaluation shown in Figure 3, the proportion of resources used was lower when processing incoming requests from end devices on the MEC server. Upon comparison between the two methods, the one lacking decision-making consumed more RAM. This increase was due to the average internal operations performed by the virtual machine before responding to the end device. The disparity in RAM consumption stems from the filtration and correspondence matching processes executed by the decision-making technology in the method integrating decision-making.

Additionally, the evaluation of storage resources in Figure 5 demonstrated that, across all instances, the average storage resources were higher in the approach without decision-making compared to the one integrating decision-making. Upon our assessment of computing and storage resources, we noted that none of the MEC server resources were fully utilized when processing end device requests in any of the scenarios. This observation validates our conclusion that the constraints specified in Equation 1 were satisfied.

B. END DEVICES' REQUESTS PROCESSING LOCATION

In Figure 6, a noticeable trend emerges: as the number of users increases, there's a simultaneous rise in the average number of requests processed by both external MEC servers and the cloud server. Similarly, there's an observed increment in the average number of tasks computed by the local MEC server. This behavior is driven by the fluctuating volume of requests from illegitimate sources, showing temporal variability over time. Moreover, the surge in requests handled by external MEC servers directly correlates with the mobility of end devices within the network. Consequently, requests that require treatment in the deception NFI, demanding higher resources, are rerouted to the cloud. This is especially impacted by the influx of illegitimate requests.

C. LATENCY EVALUATION

Our method demonstrates significantly lower latency compared to the MEC server's internal architecture without a decision-making system, as illustrated in Figure 7. This reduction in latency offers a twofold advantage. Firstly, it results in notably faster response times—a critical factor for real-time applications like video conferencing and autonomous vehicles, where rapid data processing is essential for both safety and user satisfaction. Secondly, the decreased latency not only enhances the overall user experience but also streamlines network efficiency, ensuring that vital data and commands promptly reach their intended destinations. Consequently, this leads to more seamless and responsive MEC services.

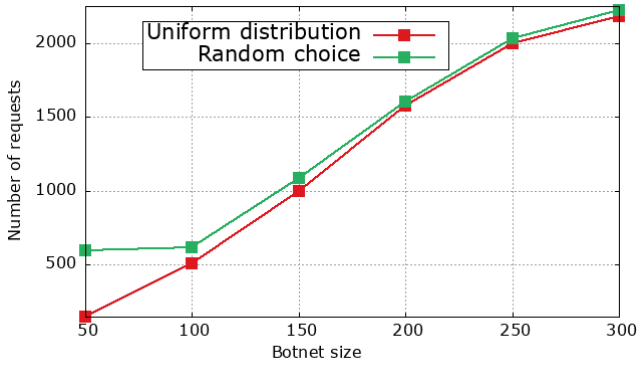


FIGURE 8. Number of request and Botnets size variation.

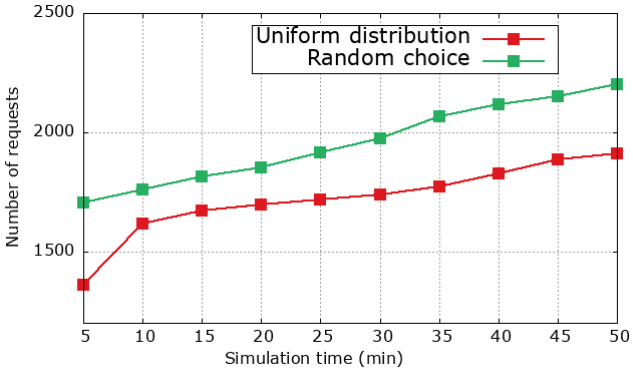


FIGURE 9. Number of requests variation over simulation time.

Furthermore, it's essential to note that these latency improvements stem from the assumption that all requests are trustworthy. In scenarios where decision-making technology is unnecessary, and there's no exchange between the latter and the SDN controller, our method excels in reducing latency. However, it's crucial to emphasize that our proposed architecture shines in scenarios involving security violations and the necessity to verify legitimacy, making it a critical solution in such contexts.

D. NUMBER OF REQUEST AND BOTNETS SIZE

Displayed in Figure 8, regardless of the botnet's size, there's a notable increase in the number of attackers' requests when utilizing the random choice method compared to the deception status code selection based on uniform distribution within the status code set. This observation suggests that the random choice strategy heightens the uncertainty associated with malicious devices, encouraging persistent request launches.

In scenarios where sufficient information about attackers is available, the decision-making system can be expanded and improved. For example, in Figure 8, when the botnet size is 100, the number of requests stored in the deception processor's database is 506 and 616 for the uniform distribution and random choice methods, respectively. This contrast highlights the effectiveness of the random choice approach in inducing uncertainty among attackers in a real-world context.

E. ARRIVAL TIME AND NUMBER OF REQUESTS

In Figure 9, we depict the dynamic correlation between the number of requests and their arrival time. This feature enables the defender to monitor the system's capability to accumulate requests as simulation time advances, offering valuable insights into the progression of request activities. Our observations consistently demonstrate an upward trend, signifying an increasing volume of requests over time. This pattern highlights the persistent engagement of malicious devices in initiating DDoS requests and their resolute pursuit of objectives, whether it involves obtaining specific data or reaching a particular request threshold.

EXPLORING REAL-WORLD APPLICATIONS AND DEPLOYMENT SCENARIO

- **Context:**
  - Student Computing Needs: The University relies on computational resources for diverse tasks conducted by students, encompassing activities like machine learning and data processing, predominantly occurring within the campus environment.
  - Server Infrastructure: Campus servers cater to student requests, operating within specific coverage zones, interconnected with Amazon Cloud's data center to supplement resources during peak demand periods.
- **Key Components:**
  - Student Devices: Including computers and smartphones, serving as endpoints for accessing computational resources.
    - Normal Devices: Representing students engaged in legitimate server usage.
    - Malicious Devices (Botnet): Signifying groups aiming to instigate DDoS attacks.
  - MEC Server Segments:
    - SDN Controller and Decision Maker: Supervising decision-making processes and request management.
    - Processor Component: Executing computation tasks.
    - Receiver and Sender: Handling incoming and outgoing requests.
    - Checker: Verifying resource availability for processing requests.
    - Deception Component: Identifying and responding to potential DDoS threats.
  - Cloud Integration: Utilized as a supplementary resource pool when local MEC server capacities are exceeded.
- **Processing Student Requests:**
  - Request Handling: Upon request receipt, the decision-making module assesses legitimacy.

TABLE 5. Comparison between our proposed work and existing ones.

Approaches	Technologies						Resources management			
	SDN	NS	NFV	SFC	Cyber deception	Decision making	Computing	Latency	Storage	Bandwidth
[15] approach	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes
[7] approach	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No
[29] approach	Yes	No	Yes	No	No	No	Yes	Yes	No	Yes
[30] approach	No	No	Yes	No	No	No	No	Yes	No	No
[31] approach	Yes	Yes	Yes	No	No	No	Yes	No	Yes	Yes
Our architecture	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

- Legitimate Request Flow:
  - Rigorous checks ensure resource adequacy; if available, the processor manages computation.
  - Post-processing, responses are relayed via the SDN controller to students, mirrored in Amazon Cloud.
- Resource Insufficiency or Coverage Gap:
  - Excess requests beyond local resources are redirected to Amazon Cloud.
  - Requests from areas beyond server coverage redirect to the nearest connected server.
- Detecting and Mitigating DDoS Threats:
  - DDoS threats detected by the decision maker redirect to the deception component.
  - The deception component formulates responses, prevents attacks, and logs details for reference.
- Deception Component's Role: Subsequent requests cross-check against the database to identify and prevent potential future attacks.

This comprehensive exploration delves into the intricate functionalities of secure MEC systems, emphasizing their adaptability in legitimate resource utilization by students and robust defense mechanisms against emerging threats.

DISCUSSION

As illustrated in Table 5, we conducted a comparative analysis between our proposed architecture and two existing ones. Significantly, our architecture distinguishes itself as the sole framework to incorporate software-defined networking, Network Function Virtualization, Service Function Chaining, Network Slicing, cyber deception, and QoS evaluation concurrently. This distinctive amalgamation of features offers insightful perspectives into the outcomes derived from our simulations, contributing significantly to a comprehensive understanding of our approach.

V. CONCLUSION

In this research, we introduced an innovative MEC architecture that integrates SDN, NFV, SFC, NS, and decision-making technologies to bolster the QoS provided to end devices. Our approach incorporates a deception VNF aimed at countering cyber threats from malicious entities. We meticulously modeled MEC server resources, including computing, storage, and bandwidth, to ensure their efficient utilization. Additionally, we developed a cyber deception framework for engaging with malicious devices, employing both uniform distribution and random selection methods.

The newly proposed MEC architecture centers around the MEC server, offering a detailed delineation of its internal components, emphasizing SDN, NFV, SFC, NS, and the decision-maker. Subsequent simulations, executed using EdgeCloudSim and HTTPS services, simulated DDoS request-response scenarios within the cyber deception framework. The simulation results unequivocally demonstrate our architecture's effective management of computing and storage resources, leading to a substantial reduction in end-to-end latency for communications. Moreover, our proposed architecture successfully upholds QoS standards while introducing a crucial filtration component to mitigate potential security threats. The cyber deception approach not only boosts the volume of collected requests from attackers but also sheds light on the evolution of request volumes over time, which could significantly enhance decision-making technology.

FUTURE DIRECTION

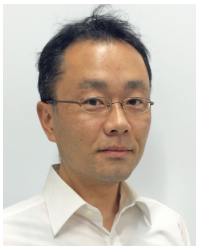
While our simulations offer promising insights, several challenges require further exploration. It would be particularly valuable to conduct a comprehensive assessment of the security risks inherent in interactions between end devices and the MEC server. Additionally, the practical implementation of our proposed architecture represents a crucial direction for our future work. Deploying this architecture within a real-world testbed environment will mark a significant step forward. Such real-world implementation will initiate a series of projects focused on MEC architectures, centered on integrating SDN, NFV, SFC, NS, and decision-making technologies. This will allow us to validate and refine the proposed framework's practical utility.

REFERENCES

[1] X. Zhang, W. Wu, S. Liu, and J. Wang, "An efficient computation offloading and resource allocation algorithm in RIS empowered MEC," *Comput. Commun.*, vol. 197, pp. 113–123, Jan. 2023.
[2] Z. Xiao, Y. Chen, H. Jiang, Z. Hu, J. C. S. Lui, G. Min, and S. Dustdar, "Resource management in UAV-assisted MEC: State-of-the-art and open challenges," *Wireless Netw.*, vol. 28, no. 7, pp. 3305–3322, Oct. 2022.
[3] Y. He, M. Yang, Z. He, and M. Guizani, "Computation offloading and resource allocation based on DT-MEC-assisted federated learning framework," *IEEE Trans. Cognit. Commun. Netw.*, vol. 9, no. 6, pp. 1707–1720, Dec. 2023.
[4] X. Lyu, H. Tian, L. Jiang, A. Vinel, S. Maharjan, S. Gjessing, and Y. Zhang, "Selective offloading in mobile edge computing for the green Internet of Things," *IEEE Netw.*, vol. 32, no. 1, pp. 54–60, Jan. 2018.

[5] B. Paharia and K. Bhushan, "A comprehensive review of distributed denial of service (DDoS) attacks in fog computing environment," in *Handbook of Computer Networks and Cyber Security: Principles and Paradigms*, 2020, pp. 493–524.
[6] M. J. A. Bonab and R. S. Kandovan, "QoS-aware resource allocation in mobile edge computing networks: Using intelligent offloading and caching strategy," *Peer-to-Peer Netw. Appl.*, vol. 15, no. 3, pp. 1328–1344, May 2022.
[7] J. Kabdjou, E. F. Tagne, D. B. Rawat, J. Acosta, and C. Kamhoua, "Cyber deception system based on Monte Carlo simulation in the mobile edge computing (MEC)," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Jul. 2023, pp. 1–7.
[8] A. Filali, A. Abouaomar, S. Cherkaoui, A. Kobbane, and M. Guizani, "Multi-access edge computing: A survey," *IEEE Access*, vol. 8, pp. 197017–197046, 2020.
[9] Y. Deng, H. Jiang, P. Cai, T. Wu, P. Zhou, B. Li, H. Lu, J. Wu, X. Chen, and K. Wang, "Resource provisioning for mitigating edge DDoS attacks in MEC-enabled SDVN," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24264–24280, Dec. 2022.
[10] Z. Kotulski, W. Niewolski, T. W. Nowak, and M. Sepczuk, "New security architecture of access control in 5G MEC," in *Proc. Int. Symp. Secur. Comput. Commun.* Singapore. Springer, Oct. 2020, pp. 77–91.
[11] M. A. Sayed, A. H. Anwar, C. Kiekintveld, and C. Kamhoua, "Honeypot allocation for cyber deception in dynamic tactical networks: A game theoretic approach," 2023, *arXiv:2308.11817*.
[12] H. Galadima, A. Seeam, and V. Ramsurrun, "Cyber deception against DDoS attack using moving target defence framework in SDN IoT-EDGE networks," in *Proc. 3rd Int. Conf. Next Gener. Comput. Appl. (NextComp)*, Oct. 2022, pp. 1–6.
[13] B. Al-Duwairi, W. Al-Kahla, M. A. AlRefai, Y. Abedalqader, A. Rawash, and R. Fahmawi, "SIEM-based detection and mitigation of IoT-botnet DDoS attacks," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 10, no. 2, p. 2182, Apr. 2020.
[14] M. S. Mahmoud, M. M. Hamdan, and U. A. Baroudi, "Secure control of cyber physical systems subject to stochastic distributed DoS and deception attacks," *Int. J. Syst. Sci.*, vol. 51, no. 9, pp. 1653–1668, Jul. 2020.
[15] M. L. F. Sindjoung, M. Velempini, and A. B. Bomgni, "A MEC architecture for a better quality of service in an autonomous vehicular network," *Comput. Netw.*, vol. 219, Dec. 2022, Art. no. 109454.
[16] M. L. F. Sindjoung, M. Velempini, and C. T. Djamegni, "A data security and privacy scheme for user quality of experience in a mobile edge computing-based network," *Array*, vol. 19, Sep. 2023, Art. no. 100304.
[17] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.
[18] H. Adhami, M. Alja'afreh, M. Hoda, J. Zhao, Y. Zhou, and A. El Saddik, "Suitability of SDN and MEC to facilitate digital twin communication over LTE—A," *Digit. Commun. Netw.*, Jun. 2023.
[19] M. L. F. Sindjoung, and P. Minet, "Estimating and predicting link quality in wireless IoT networks," *Ann. Telecommun.*, vol. 77, pp. 253–265, 2021, doi: 10.1007/s12243-021-00835-1.
[20] V. Jain, S. Aggarwal, S. Mehta, and R. Hebbalaguppe, "Synthetic video generation for robust hand gesture recognition in augmented reality applications," 2019, *arXiv:1911.01320*.
[21] Q. Liao, H. Li, S. Kang, and C. Liu, "Application layer DDoS attack detection using cluster with label based on sparse vector decomposition and rhythm matching," *Secur. Commun. Netw.*, vol. 8, no. 17, pp. 3111–3120, Nov. 2015.
[22] G. Mirjalily and Z. Luo, "Optimal network function virtualization and service function chaining: A survey," *Chin. J. Electron.*, vol. 27, no. 4, pp. 704–717, Jul. 2018.
[23] G. Davoli, W. Cerroni, C. Contoli, F. Foresta, and F. Callegati, "Implementation of service function chaining control plane through OpenFlow," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2017, pp. 1–4.
[24] L. U. Khan, I. Yaqoob, N. H. Tran, Z. Han, and C. S. Hong, "Network slicing: Recent advances, taxonomy, requirements, and open research challenges," *IEEE Access*, vol. 8, pp. 36009–36028, 2020.
[25] S. Zhang, W. Quan, J. Li, W. Shi, P. Yang, and X. Shen, "Air-ground integrated vehicular network slicing with content pushing and caching," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2114–2127, Sep. 2018.
[26] M. L. F. Sindjoung and P. Minet, "Wireless link quality prediction in IoT networks," in *Proc. 8th Int. Conf. Perform. Eval. Model. Wired Wireless Netw. (PEMWN)*, Nov. 2019, pp. 1–6.
[27] P. Aggarwal, C. Gonzalez, and V. Dutt, "Looking from the hacker's perspective: Role of deceptive strategies in cyber security," in *Proc. Int. Conf. Cyber Situational Awareness, Data Anal. Assessment (CyberSA)*, Jun. 2016, pp. 1–6.
[28] W. Steingartner, D. Galinec, and A. Kozina, "Threat defense: Cyber deception approach and education for resilience in hybrid threats model," *Symmetry*, vol. 13, no. 4, p. 597, Apr. 2021.
[29] L. Yala, P. A. Frangoudis, and A. Ksentini, "Latency and availability driven VNF placement in a MEC-NFV environment," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.
[30] R. Cziva, C. Anagnostopoulos, and D. P. Pezaros, "Dynamic, latency-optimal vNF placement at the network edge," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, Apr. 2018, pp. 693–701.
[31] H. Peng, Q. Ye, and X. S. Shen, "SDN-based resource management for autonomous vehicular networks: A multi-access edge computing approach," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 156–162, Aug. 2019.

JOELLE KABDJOU received the master's degree in networks and distributed services from the Faculty of Science, University of Dschang, in July 2021. She is currently pursuing the Ph.D. degree with Soka University, Tokyo, Japan. Her current research interests include software-defined networking, service function chains, network slicing, network function virtualization, deep reinforcement learning, cyber security, mobile edge computing, and machine learning. During the graduate studies, she was ranked first in her department.



NORIHIKO SHINOMIYA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in information systems science from Soka University, Tokyo, Japan, in 1995, 1997, and 2001, respectively. In 2000, he joined Network Systems Laboratories, Fujitsu Laboratories Ltd., as a Research Engineer, where he engaged in development of a planning and design tool for wavelength division multiplexing (WDM) networks and a control method of IP networks. Since 2005, he has been with the Department of Information Systems Science, Faculty of Engineering, Soka University. He has been a Professor and engaged in the research and development of design method, control architecture, and management system based on the graph theoretical algorithms. He is a member of IEEE CAS, ComSoc, and Computer societies. He received the five best paper awards at international conferences, including IEEE ICUMT in 2010, IARIA ICONS (two papers) in 2014, and IARIA ICN in 2015 and 2016. He served as a Secretary for the IEEE CAS Society Japan Chapter, from 2012 to 2013.